Predict and Natural Development Server

General Information

Natural Development Server (NDV) stores the following information into Predict:

- Structure of Application Descriptions (APD)
- Locks

Note:
Locks are internal used objects only and can not be maintained by Predict.

Note:
For more information about Applications see **Introducing Natural's Single Point of Development**.

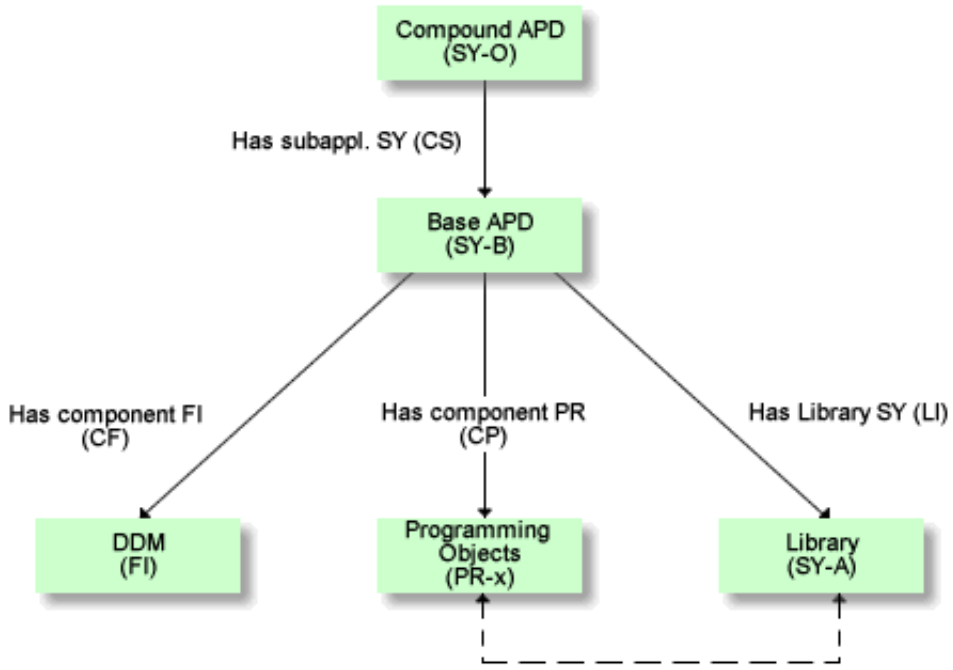
This section covers the following topics:

- Documenting Natural Development Server in Predict
- Documenting Base Application Descriptions
- Documenting Compound Application Descriptions
- Documenting Data Definition Modules (DDM)
- Documenting Natural Programming Objects
- Documenting Libraries

Documenting Natural Development Server in Predict

The following NDV objects can be documented in Predict:

- Base Applications Descriptions
- Compound Applications Descriptions
- Data Definition Modules
- Natural Programming Objects
- Libraries
- Message files



The following table provides an overview of how different NDV objects are documented.

Natural Development Server	Documented in Predict with
Base APD	System object of type B
Compound APD	System object of type O
DDM	File object
Natural Programming Objects	Program object of corresponding type
Library	System object of type A
Message file	Program object of type 2

Documenting Base Application Descriptions

Base Application Descriptions are documented as objects of type System with system type B.
Base Application Descriptions have the following specific attributes:

- Server name
- Port
- Profile name
- Profile DBnr
- Profile Fnr

For Base APDs the following specific associations exists:

- *Has component FI* with association code CF:
  which Data Definition Modelues belongs to this Application description.
- *Has component PR* with association code CP:
  which Natural Programming Objects belongs to this Application description.
- *Has library SY* with association code LI:
  each library of Natural Programming Objects, that are linked to the Base APD with association *Has component PR*, must be linked to the Base APD.

Note:
The association *Has library SY* is build automatically when changing the association *Has component PR*.

Documenting Compound Applications Descriptions

Compound Application Descriptions are documented as objects of type System with system type O.
Compound Application Descriptions have no specific attributes.

The association from Compound APD to Base APD is named *Has subappl. SY* with association code CS.

Note:
Predict Maintenance functions ensure that only Systems of system type B are linked to Compound APDs.

Documenting Data Definition Modules (DDM)

Data Definition Modules are documented as objects of type File.
The association from a Base APD to the Data Definition Modules is named *Has component FI* with asssociation code CF.

Documenting Natural Programming Objects

Natural Programming Objects are documented as objects of type Program with corresponding program type.
The association from a Base APD to the Natural Programming Objects is named *Has component PR* with asssociation code CP.

The following rules apply for Natural Programming Objects linked to a Base APD:

- The implementation pointer must be full qualified.
- All members must be on the same Natural System File.
- For each Natural Programming Object the library this object is in must be documented as a System of system type A with the association *Has library SY* (association code LI).
- If database number or file number of the implementation pointer is changed, the Natural Programming Object is removed from the Base APD. If it is also the last member with the library in this Base APD, the corresponding system type A, representing the library, must be removed from association *Has library SY* too.

Documenting Libraries

Libraries are documented as objects of type System with system type A.
The association from a Base APD to the libraries is named *Has library SY* with asssociation code LI.

